Principles of Safe Autonomy ECE 484
Fall 2022 Lecture 1
Professor Sayan Mitra (course director)
Professor Ujjal Bhowmik (your instructor for the semester)
Aug 23, 2022
https://publish.illinois.edu/safe-autonomy/
Welcome from Safe Autonomy team!

Professor
Ujjal Bhowmik (ubhowmik)

Graduate TAs
Yan Miao (yanmiao2)
Peter Du (peterdu2)
Hongyi Li (hli106)

UG Lab Assistants / graders
Raj Joshi
Plan for today

► What is this course about?
► How will this course work?
► Introduction to safety
Autonomous systems will be awesome!

Driverless cars will make us more productive
- Average American drives 13,474 miles (300 hrs) per year

Our cities will be greener
- 40% of city surface is parking

Travel and deliveries will be safer
- 32K+ fatalities and 3M+ injuries every year
100 years of progress in safer roads

1. Traffic infrastructure (e.g., lane markings, traffic signals, ...)
2. Police enforcement and traffic regulations
3. Driver training
4. Passenger safety (e.g., seatbelts, airbags)
5. Improved vehicle design (e.g., crumple zones)
6. Rear-view and blind spot sensors (e.g., camera)
7. Advanced Driver Assistance Systems (e.g., ABS, ACC, etc.)
The years in blue indicate when the industry leaders thought these predictions would come to pass. I have highlighted all the dates up through 2021, now numbering 17 of the 23 predictions. Not one of them has happened or is even close to happening.

**FORECASTS:**

- NVIDIA to introduce level-4 enabling system by 2018 (2017)
- NuTonomy to provide self-driving taxi services in Singapore by 2018, expand to 10 cities around world by 2020 (2016)
- Delphi and MobilEye to provide off-the-shelf self-driving system by 2019 (2016)
- Ford CEO announces fully autonomous vehicles for mobility services by 2021 (2016)
- Volkswagen expects first self-driving cars on the market by 2019 (2016)
- GM: Autonomous cars could be deployed by 2020 or sooner (2016)
- BMW to launch autonomous iNext in 2021 (2016)
- Ford’s head of product development: autonomous vehicle on the market by 2020 (2016)
- Baidu’s Chief Scientist expects large number of self-driving cars on the road by 2019 (2016)
- First autonomous Toyota to be available in 2020 (2015)
- Elon Musk now expects first fully autonomous Tesla by 2018, approved by 2021 (2015)
- US Sec Trans: Driverless cars will be in use all over the world by 2025 (2015)
- Uber fleet to be driverless by 2030 (2015)
- Ford CEO expects fully autonomous cars by 2020 (2015)
- Next generation Audi A8 capable of fully autonomous driving in 2017 (2014)
- Jaguar and Land-Rover to provide fully autonomous cars by 2024 says Director of Research and Technology (2014)
- Fully autonomous vehicles could be ready by 2025, predicts Daimler chairman (2014)
- Nissan to provide fully autonomous vehicles by 2020 (2013)
- Truly autonomous cars to populate roads by 2028-2032 estimates insurance think tank executive (2013)
- Continental to make fully autonomous driving a reality by 2025 (2012)

Hubris on a mass delusion scale. Audi fully autonomous by 2017? That is Teslan in its delusion level.
Building autonomous cars will be harder than going to the Moon.

Some of you will be working on this problem for decades!
Another transportation revolution from a century ago
1902: after thousands of tests Oliver and Wilbur Wright had acquired the knowledge and the skill to fly. *They could soar, float, dive, circle and land, all with assurance.* Now they had only to build a motor. - David McCullough
Goal: understand the components and discover the principles for building reliable autonomy from unreliable components
Autonomous GEM vehicle: An example CPS

- **Perception**
- **Planning and decision**
- **Control**
- **Vehicle dynamics**

**Environment**

- Sensing
- Perception
- Decisions and planning
- Control

**env**: Lidar / vision

**Sensing**
- Physics-based models of cameras, LIDAR, radar, GPS, and so on.

**Perception**
- Programs for object tracking, scene understanding, and so on.

**Decisions and planning**
- Programs and multi-agent models of pedestrians, cars, and so on.

**Control**
- Dynamical models of vehicle engine, powertrain, steering, tires, and so on.
Plan for today

- What is this course about?
- How will this course work?
- Introduction to safety
Why are we here? Course goals

**Know**
- Components of an autonomous system, safety standards, ...
- How to use software modules for perception, planning, control, ROS, Yolo, OpenCV, Z3, ...

**Do**
- Code and analyze algorithms for perception clustering, convolution, filtering, edge detection, filtering, localization, planning, formal verification
- Plan, propose, organize and execute a team project

**Understand**
- Models, algorithms, data, biases, assumptions for building trustworthy autonomous systems
- Theoretical properties of algorithms and their limitations

**Get inspired**
- Become the Isaac Newton of Autonomy

“To do things right, first you need love, then technique.” – Antoni Gaudí
Course structure

- **Perception**
  - Planning and decision

- **Control**
  - ODE, Stability, PID

- **Vehicle dynamics**

- **Sensing**

- **Environment**
  - Lidar / vision

- **Safety, end-to-end testing, simulation, system integration**
  - MP0, MP5, Project
  - State machines, model checking, hypothesis testing, ROS

- **Convolution**
  - Bayes filter
  - SLAM

- **Vehicle dynamics**

- **MP2 Project**
  - Decisions and planning

- **MP3 Project**
  - Sensing

- **MP1 Project**
  - Perception

- **MP4 Project**
  - Control

- **MP0 Project**
  - Safety, end-to-end testing, simulation, system integration
484 objectives & overview

- Learn architectures and fundamental building blocks of autonomy
  - Perception, planning, control, verification
- Gain hands-on experience in building an end-to-end system with your team
- Lectures, MPs, HWs, Labs, Exams, Project

Diagram:
- Perception → Planning and decision → Control → Vehicle dynamics
  - Sensing
  - Environment
  - env: Lidar / vision

Languages and Tools:
- Python
- CARLA
- Gazebo
- ROS
- OpenCV
- YOLO
- Python Language
About the course

Everything starts here: [https://publish.illinois.edu/safe-autonomy/](https://publish.illinois.edu/safe-autonomy/)

- Schedule, lab, resources, papers, homework, MP, code, project, gitlab links

Campuswire for announcements, but no SLA

- Discussions, forming teams, occasional polls, feedback
- Do not expect to get answers to HWs, MPs, and exam related questions in the last minute.

Canvass for MP release and grades
Schedule:  [https://publish.illinois.edu/safe-autonomy/](https://publish.illinois.edu/safe-autonomy/)

- Simple safety (MP0, 1 week)
- Perception (MP1 lane detection, 2 weeks)
- Modeling and control (MP2 vehicle control, 2 weeks)
- Filtering: localization particle filtering (MP3 localization, 2 weeks)
- Midterm 1
- Planning (MP4 planning, 2 weeks)
- Fall Break
- Decision making RL (MP5 racing, end-to-end safety)
- End-to-end system and safety analysis
- Midterm 2

- Group formation
- Lab safety training
- Labs, MPs
- Project pitch
- Fall Break
- Intermediate checking
- Practice presentation
- Final presentation
Course materials
Lecture notes, slides, code, video lectures, lab manuals created and curated from recent research publications

Course reader

Reference books:
- Probabilistic robotics, By Sebastian Thrun, Wolfram Burgard and Dieter Fox, 2005
- Verifying Cyber-physical Systems, Sayan Mitra, MIT Press 2021
Course: components and (tentative) weights

- 5-6 programming assignments or MPs 45% (group)
  - ROS + Python, Ubuntu, VM BYOD or use lab workstations
  - labs (Friday 4-8 pm starting tomorrow)
  - Office hours

- Homework assignments 10% (individual)
  - math, analysis, critical reasoning; preparation for midterms

- Midterms x2 20% (individual)

- Mini project 20% (group): more on this later, 2 tracks:
  - A. Dev and test concepts on GEM
  - B. GRAIC autonomous racing competition / testing

- Participation 5%
  - Lab and class attendance and participation, early feedback on notes, solve exercise problems in class notes, class participation, MP beta testers

<table>
<thead>
<tr>
<th>Tentative grade boundaries</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
</tbody>
</table>
Homework, participation, & exam: Individual work 35%

- Testing principles, concepts
- Read course notes and slides routinely; exercises are provided
- Homework sets (synchronized with MPs)
- 2 in-class midterms TBA
- No final exam
Teamwork: MP, labs, and mini project

- In groups: Form your group of 3-4 now! Make your group (form), make new friends
  - If you do not have a group by ??? we will assign you a group
- Each MP will build a significant component of an autonomous system over 2 weeks
- Use our VM or your own computer with Ubuntu 20 or broadband internet connection to remote login to lab computers
- **Attend your lab section this Friday ECEB 5072**
  - MP walkthrough, setup, bridge the lecture and the assignments
- **MP0+HW0 released (8/22) due (9/9)**
Mini projects: explore, inspire, and impress

**Hardware Track.** Build on existing SW to live demo at Highbay
- Reliable parallel parking, lane following and pedestrian avoidance outdoor track, biker intent estimation and reaction
- COVID related uncertainty about highbay access; Frontload the work

**Software Track.** Participate in an open simulation-based autonomous racing competition
- Build your own driver software, test it in a number of environments
- Automatically create complex scenarios

**Outcomes:** Write research papers, jumpstart grad research, career in autonomy, incubate startup ideas, sharpen presentation skills

We provide: Polaris GEM vehicle (camera, LIDAR, RADAR, IMU, GPS, and drive-by-wire system) modules for pedestrian detection, lane tracking, and vehicle control, a vehicle simulator, and testing facility (highbay) with indoor positioning system. GRAIC autonomy software stack

**Expertise (TA, lab and office hours, TBD)**

**Timeline:** *Get started, be a member of IRL from this link*
- High-bay virtual site visit and training (in next 2 weeks)
- Project pitch before Fall break
- **Public presentation, demo, awards (End of semester)**
Software projects

- Participate in **GRAIC autonomous** racing competition (co-located with CPSWeek in 2022)
- Dozens of vehicles, tracks, scenarios, and other participants
- [https://popgri.github.io/Race/](https://popgri.github.io/Race/)
Hardware projects

- Design, deploy, and test autonomy capabilities like autosteer, autopilot, lane-assist, emergency braking on the GEM EV platform

- See final presentation videos here: https://tinyurl.com/484spring22
Absolute safety analysis
A popular position about autonomy

“Collect lots of data, build faster computers, and train larger neural networks, and safe autonomous vehicles will follow” --- A founder

How to assure safety of an AV? Run tests

In an absolute sense:

“Testing can be used to show the presence of bugs, but never to show their absence!” --- Edsger W. Dijkstra

Because there are infinitely many executions and we can only test finitely many of those in any testing algorithm

In a probabilistic sense also, purely using data to gain safety assurance is not practical
Naively collecting test driving miles is also not going to work

Probability of a fatality caused by an accident per one hour of human driving is known to be $10^{-6}$

Assume* that for AV this has to be $10^{-9}$

Data required to guarantee a probability of $10^{-9}$ fatality per hour of driving is proportional to its inverse, $10^{9}$ hours, 30 billion miles

Multi-agent, open system, with human interactions => cannot be simulated offline to generate data

Any change is software means tests have to be rerun

To learn or extrapolate about all---infinitely many---executions from a finite sampling of executions, we need to make some assumptions about the system. A collection of these assumptions defines a model

Different types of model (and data) for sensing, control, planning, and we need to understand how to analyze and compose them

*On a Formal Model of Safe and Scalable Self-driving Cars by Shai Shalev-Shwartz, Shaked Shammah, Amnon Shashua, 2017 (Responsibility Sensitive Safety)*
Autonomous GEM vehicle: An example CPS

Sensing
- Physics-based models of cameras, LIDAR, radar, GPS, and so on.

Perception
- Programs for object tracking, scene understanding, and so on.

Decisions and planning
- Programs and multi-agent models of pedestrians, cars, and so on.

Control
- Dynamical models of vehicle engine, powertrain, steering, tires, and so on.

env: Lidar / vision

Environment

Perception → Planning and decision → Control → Vehicle dynamics

Sensing
Next: How can we use a simple model to get absolute safety guarantees

► A simple class of models: automata
► What are executions of automata?
► What are safety requirements?
► Reachable states, Invariants for safety guarantees
A “simple” safety scenario

A car moving down a straight road has to detect any pedestrian (or another car) in front of it and stop before it collides.

Automatic Emergency Braking

Not a trivial requirement

Today: There is no enforced standard for testing AEB
“simple” ≠ Easy
Modeling the scenario

► What is a model of a system?

► A mathematical model describes how a system behaves.
  ➤ What are the key parameters and states?
  ➤ How are the parameters selected by nature?
  ➤ What are the initial conditions of the state?
  ➤ How do the state change over time? ...
  ➤ What parts of the model are available for observation/analysis?

► Models include the implicit and explicit assumptions (biases) we are making about the system
Model (switch to notes)
“All models are wrong, some are useful.”
MP0: Simulate model for testing
An honest scientific approach

1. Create detailed *mathematical models* of the autonomous systems and its environment
2. Enumerate the precise *requirements* of the system and the conditions on the environment under which it is supposed to work
3. Analyze the system to either
   - *prove* that all behaviors meet the requirement (perhaps with high probability)
   - find counter-examples, corner cases, etc., debug and repeat

- Currently there are fundamental flaws in making this work for autonomous systems

- **Why study this approach?**
  - Careful reasoning can expose flawed assumptions, bad design choices
  - The approach has been successful in other industries: microprocessors, aviation, cloud computing, nuclear, ..
  - Big strides in the last few years
  - Working deliberately towards a more perfect understanding is a worthwhile intellectual struggle